

DEVICE AND METHOD FOR IMAGE DATA PROCESSING

Patent number: JP2001257886

Publication date: 2001-09-21

Inventor: SHIMODA AKIRA

Applicant: CANON INC

Classification:


- international: H04N1/409; B41J2/485; G06T5/20;
G09G5/36; H04N1/21; H04N9/68

- european:

Application number: JP20000065568 20000309

Priority number(s):

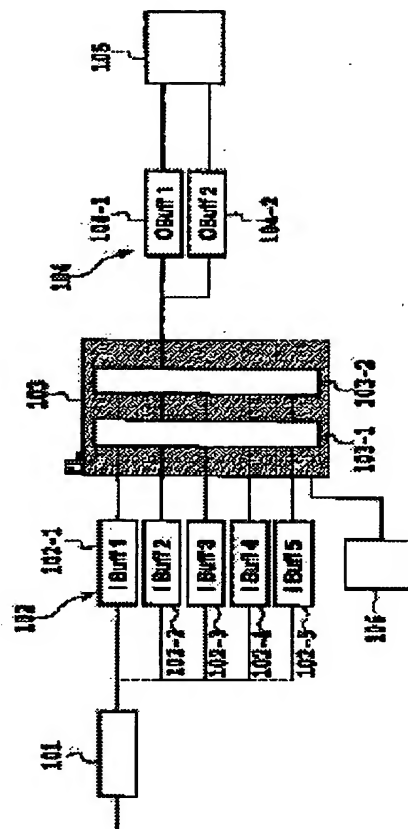
Also published as:

 JP2001257886 (A)

Abstract of JP2001257886

PROBLEM TO BE SOLVED: To shorten an image data processing time.

SOLUTION: A filter block 103 reads the leftmost data out of five operation input buffers 102-1 to 102-5 at the same time. Then adjacent data are read out at the same time and thus data are read out in order. At the start of operation, the readout operation is repeated five times successively to set 25 pieces of data. A filter block 103 performs 1st operation and stores its result in an out buffer 104-1. This operation is performed continuously for one raster. Data used for the arithmetic processing are used through single-time readout operation in sequence, so the readout time of the data processing is shortened to efficiently perform the image processing.



THIS PAGE BLANK (USPTO)

(19)日本国特許庁(JP)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開2001-257886

(P2001-257886A)

(43)公開日 平成13年9月21日(2001.9.21)

(51)Int. Cl. ⁷	識別記号		F I		テーマコード(参考)	
H 0 4 N	1/409		G 0 6 T	5/20	B	2C062
B 4 1 J	2/485				C	5B057
G 0 6 T	5/20		H 0 4 N	1/21		5C066
				9/68	1 0 3	Z 5C073
G 0 9 G	5/36			1/40	1 0 1	D 5C077
審査請求 未請求		請求項の数 1 4	O L	(全 7 頁)		最終頁に続く

(21)出願番号 特願2000-65568(P2000-65568)

(22)出願日 平成12年3月9日(2000.3.9)

(71)出願人 000001007

キャノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 霜田 明良

東京都大田区下丸子3丁目30番2号 キャノ
ン株式会社内

(74)代理人 100077481

弁理士 谷 義一 (外1名)

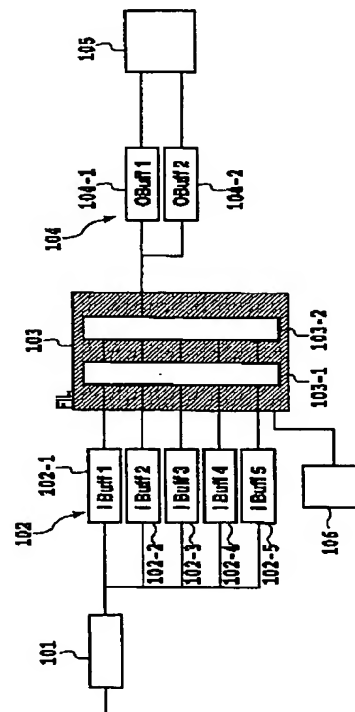
最終頁に続く

(54)【発明の名称】 画像データ処理装置および方法

(57)【要約】

【課題】 画像データ処理時間を短縮すること。

【解決手段】 フィルターブロック103は5個の演算入力バッファ102-1から102-5の各々の一番左側のデータを同時に読み出す。ついで、その隣りのデータを同時に読み出すというように順次読み出す。演算開始時にはこの読み出し動作を連続して5回行うことで25個のデータがセットされる。フィルターブロック103では、最初の演算を行い結果をアウトバッファ104-1に格納する。この動作を1ラスタについて連続して行う。演算処理において使用するデータを逐次一回の読み出し動作で使用するのでデータ処理における読み出し時間を短縮し画像処理を効率よく行うことが出来る。



【特許請求の範囲】

【請求項1】 画素データを各ライン毎に一時記憶する n 個の記憶手段と、
前記 n 個の記憶手段の各々から各ラインの画素データを並行して読み出し、読み出した $n \times m$ (m : 各ラインにおける画素数) 個のマトリックスデータ毎に演算処理を行う演算手段とを備えたことを特徴とする画像データ処理装置。

【請求項2】 請求項1において、
前記演算手段における演算出力対象画素は、 $n \times m$ のマトリックスの中心の画素であることを特徴とする画像データ処理装置。

【請求項3】 請求項1または2において、
 $n+1$ ライン目の画素データを一時記憶する記憶手段をさらに有し、
前記演算手段は、前記 $n+1$ 個の記憶手段のうちの n 個の記憶手段の各々から各ラインの画素データを並行して読み出すことを特徴とする画像データ処理装置。

【請求項4】 請求項1～3のいずれかにおいて、
前記画素データは、同一画像位置の複数の色情報を含むことを特徴とする画像データ処理装置。

【請求項5】 請求項1～4のいずれかにおいて、
前記演算手段の演算出力データを1ラインずつ順次記憶する2個以上のバッファをさらに有することを特徴とする画像データ処理装置。

【請求項6】 請求項1～5のいずれかにおいて、
前記演算手段は、演算出力対象画素データに対してエッジ強調処理を施すことを特徴とする画像データ処理装置。

【請求項7】 請求項1～5のいずれかにおいて、
前記演算手段は、演算出力対象画素データに対してスムージング処理を施すことを特徴とする画像データ処理装置。

【請求項8】 画素データを各ライン毎に n 個の記憶手段の各々に一時記憶し、
前記 n 個の記憶手段の各々から各ラインの画素データを並行して読み出し、
読み出した $n \times m$ (m : 各ラインにおける画素数) 個のマトリックスデータ毎に演算処理を行うことを特徴とする画像データ処理方法。

【請求項9】 請求項8において、
前記演算における演算出力対象画素は、 $n \times m$ のマトリックスの中心の画素であることを特徴とする画像データ処理方法。

【請求項10】 請求項8または9において、
 $n+1$ ライン目の画素データをさらに別の記憶手段に一時記憶し、
前記演算は、前記 $n+1$ 個の記憶手段のうちの n 個の記憶手段の各々から各ラインの画素データを並行して読み出して行うことを特徴とする画像データ処理方法。

【請求項11】 請求項8～10のいずれかにおいて、
前記画素データは、同一画像位置の複数の色情報を含むことを特徴とする画像データ処理方法。

【請求項12】 請求項8～11のいずれかにおいて、
前記演算出力データを1ラインずつ2個以上のバッファに順次記憶させることを特徴とする画像データ処理方法。

【請求項13】 請求項8～12のいずれかにおいて、
前記演算は、演算出力対象画素データに対するエッジ強調処理であることを特徴とする画像データ処理方法。

【請求項14】 請求項8～12のいずれかにおいて、
前記演算は、演算出力対象画素データに対するスムージング処理であることを特徴とする画像データ処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は画像データに対して演算等のデータ処理を行うための画像データ処理装置及び方法に関する。

【0002】

【従来の技術】 スキャナ等から読み込まれた画像データをプリントアウトして使用する場合等には、その使用用途によって特定の効果を持たせたり、読みとり素子の特性を補う、読みとりに際して混入したノイズデータを軽減する、などの目的で各種のデータ処理が施される。たとえば文字データが背景と重なって配置された画像データでは文字と背景の境界がぼやけていたら文字として認識しにくい。

【0003】 このように文字境界付近の輪郭が何らかの理由によりぼやけてしまった画像に対しては、これを修正する目的でエッジを強調するような画像処理を施すことがある。このような処理を行う際には文字周辺の画素データからエッジを判断し演算処理によって画像処理が施される。またエッジ強調のみならず曲線部をよりなめらかに表示する際にも周囲のデータから画像の一部を修正しスムージングを施すような画像処理が施される場合もある。

【0004】 これらの画像処理はその目的から対象画素の周囲の状況によって実際の処理結果が異なる、つまり対象画素に対してその周囲の画素データを用いて演算処理を行うことが必要となる。

【0005】 上述のように、対象画素について周囲の画素データにもとづいて処理を行おうとする場合には、それら必要なデータを順次読み出し、すべてのデータがそろった時点で演算処理を行い処理結果を出力する動作を繰り返す。たとえば演算に必要な画像データのラスタ数 n 、幅方向の画素数を m とした場合、ある画素について対象画素を含む周囲 $n \times m$ 個のデータが必要であるから、処理以前に少なくとも $n \times m$ 回の読み出し動作を行う必要がある。

【0006】 画像の左はじから画像幅方向へ向かって順

次データ処理を行っていく場合を考えると、最初に $n \times m$ 回のデータ読み出しを行い、そのデータを用いて最初の 1 画素に対する演算が行われる。次に右隣の 1 画素に対する演算を行うためには、少なくとも新たに n 個のデータを読み込む必要がある。

【0007】

【発明が解決しようとする課題】近年の PC (パーソナルコンピュータ) の高速化によりカラー画像を容易に扱うことが可能となるなど処理しなければならないデータとその速度に対する要求は益々大きくなり、よりきれいな画像表現のために解像度が精細になり、さらにはよりきめ細かい階調表現を行うなど表現し得る画質向上のために同一面積の画像データの容量は加速度的に大きくなる傾向にある。

【0008】そのためこれらのデータ処理もデータ容量に比例して増加し処理時間も増大してしまうという問題がある。

【0009】画像データは何らかの記憶媒体上におかれている。そのため、特に前述の様なデータ処理の場合、処理しようとする画素データは周囲の画素との関係で演算処理が行われる。目的とする演算処理のためには対象となる画素データはもちろん処理に必要な周囲のデータを読み出すという作業も必要になる。

【0010】よって、それらの必要な画像データを読み出した上で演算処理を行い、その処理結果を再び記憶媒体上に書き込む手順をとる。

【0011】前述のようにデータ量が大きくなれば画像データ処理のためのデータ読み出し、演算処理、処理済みデータの再書き込みの負荷はデータ量に比例して大きくなる。

【0012】そのため、これらの処理データの一つずつ順次読みだす方法では処理時間が膨大になってしまう等の問題がある。さらに画像処理の多くは処理そのものを目的として専用の機器において行われるわけではなく、たとえば印刷記録のために画像記録装置の一部において行われたりディスプレイへの表示のために映像装置の一部などその他の機能と同時に実現する必要があることが一般的である。

【0013】これらの機器においては前述の画像処理に必要なデータの読み出しや書き込みは他の処理に用いるバスを共用して行われデータ処理だけに処理能力のすべてを使用できるわけではない。

【0014】そのため、画像処理負荷の増大が装置全体のパフォーマンスに及ぼす影響を考えても特定の処理は速やかに行う必要がある。

【0015】そこで本発明の目的は、以上のような問題を解消した画像データ処理装置および方法を提供することにある。

【0016】

【課題を解決するための手段】上記目的を達成するた

め、請求項 1 の発明は、画素データを各ライン毎に一時記憶する n 個の記憶手段と、前記 n 個の記憶手段の各々から各ラインの画素データを並行して読み出し、読み出した $n \times m$ (m : 各ラインにおける画素数) 個のマトリックスデータ毎に演算処理を行う演算手段とを備えたことを特徴とする。

【0017】また請求項 2 の発明は、請求項 1 において、前記演算手段における演算出力対象画素は、 $n \times m$ のマトリックスの中心の画素であることを特徴とする。

【0018】さらに請求項 3 の発明は、請求項 1 または 2 において、 $n+1$ ライン目の画素データを一時記憶する記憶手段をさらに有し、前記演算手段は、前記 $n+1$ 個の記憶手段のうちの n 個の記憶手段の各々から各ラインの画素データを並行して読み出すことを特徴とする。

【0019】さらに請求項 4 の発明は、請求項 1~3 のいずれかにおいて、前記画素データは、同一画像位置の複数の色情報を含むことを特徴とする。

【0020】さらに請求項 5 の発明は、請求項 1~4 のいずれかにおいて、前記演算手段の演算出力データを 1 ラインずつ順次記憶する 2 個以上のバッファをさらに有することを特徴とする。

【0021】さらに請求項 6 の発明は、請求項 1~5 のいずれかにおいて、前記演算手段は、演算出力対象画素データに対してエッジ強調処理を施すことを特徴とする。

【0022】さらに請求項 7 の発明は、請求項 1~5 のいずれかにおいて、前記演算手段は、演算出力対象画素データに対してスムージング処理を施すことを特徴とする。

【0023】さらに請求項 8 の発明は、画素データを各ライン毎に n 個の記憶手段の各々に一時記憶し、前記 n 個の記憶手段の各々から各ラインの画素データを並行して読み出し、読み出した $n \times m$ (m : 各ラインにおける画素数) 個のマトリックスデータ毎に演算処理を行うことを特徴とする。

【0024】さらに請求項 9 の発明は、請求項 8 において、前記演算における演算出力対象画素は、 $n \times m$ のマトリックスの中心の画素であることを特徴とする。

【0025】さらに請求項 10 の発明は、請求項 8 または 9 において、 $n+1$ ライン目の画素データをさらに別の記憶手段に一時記憶し、前記演算は、前記 $n+1$ 個の記憶手段のうちの n 個の記憶手段の各々から各ラインの画素データを並行して読み出して行うことを特徴とする。

【0026】さらに請求項 11 の発明は、請求項 8~10 のいずれかにおいて、前記画素データは、同一画像位置の複数の色情報を含むことを特徴とする。

【0027】さらに請求項 12 の発明は、請求項 8~11 のいずれかにおいて、前記演算出力データを 1 ラインずつ 2 個以上のバッファに順次記憶させることを特徴と

する。

【0028】さらに請求項13の発明は、請求項8～12のいずれかにおいて、前記演算は、演算出力対象画素データに対するエッジ強調処理であることを特徴とする。

【0029】さらに請求項14の発明は、請求項8～12のいずれかにおいて、前記演算は、演算出力対象画素データに対するスムージング処理であることを特徴とする。

【0030】

【発明の実施の形態】（実施例1）以下に図面を参照して本考案の実施例の説明を進める。

【0031】本実施例ではスキャナーから取り込んだ画像データに対してエッジ強調のためにフィルター処理を施す回路を例に説明を行う。

【0032】図2は本実施例における画像データの演算処理入力データ範囲である。ここに並んだ丸は各々画素を示す。201は演算対象になる画素である。202は本実施例における演算のデータ入力範囲で、図示するように25個の画素データを用いて演算処理を行う。

【0033】203は演算処理対象画素、204は処理済みの画素データの出力範囲を示す。この図面からわかるように出力データ範囲に対して入力データは上下左右方向に2ドットづつ余分に与えられている。

【0034】すなわち演算に必要な画素データが $n \times m$ 個で、 $n = m$ でかつ n, m が奇数である場合、 $a = (n - 1) / 2$

で表される a 個の入力画素を出力画素数に対して上下左右に余分に与える。

【0035】図3は図2の202で囲った範囲の画素、つまり演算処理に用いる画素にそれぞれ名称をつけたものである。ここに示す各画素 a_{11} から a_{55} は図4に示す様にR、G、Bの各色の濃度が8ビットで表された24ビットの画素データである。

【0036】先の説明から明らかなようにこの図3においては、 a_{33} が演算対象になる画素である。

【0037】次に画素に対する演算処理の内容について例をあげて説明する。本実施例における演算回路は注目画素すなわち処理対象画素に対して、周囲画素のデータの状況から各々画素の濃度を加減する。

【0038】画像全体に対して順次同様の処理を施すことでたとえばエッジ強調効果を得る。演算処理によってエッジ強調効果を得る原理については数学的手法によって説明されるが本発明の主題とは直接関係ないのでここでは省略し、処理手順についてのみ説明を行う。

【0039】演算処理はたとえば以下の式で表され、処理対象画素に対して周囲画素の状況が反映された出力を得る。

【0040】出力画素の値をXとすると

$$X = a_{33} \times A + (a_{22} + a_{23} + a_{24} + a_{32} +$$

$$a_{34} + a_{42} + a_{43} + a_{44}) \times B + (a_{11} + a_{12} + a_{13} + a_{14} + a_{15} + a_{25} + a_{35} + a_{45}) \times C + (a_{21} + a_{31} + a_{41} + a_{51} + a_{52} + a_{53} + a_{54} + a_{55}) \times D$$

（ここで、A、B、C、Dは正負の係数である。）で、処理内容によって、A、B、C、Dの値を変えて、期待する画像処理効果を得る。

【0041】図1は本実施例における画像処理演算回路周辺のブロック図である。101は人カマスクブロック、102は演算に必要な画像データを蓄える演算入力バッファであって、本実施例では102-1から102-5の5個の演算入力バッファをもつ。

【0042】103はフィルターブロックであって、演算入力バッファ102に蓄えられたデータを順次読み出した上で演算処理を行い、出力バッファ104に処理結果を格納する。103-2は演算回路、103-1は演算回路103-2へ演算入力バッファ102-1から102-5のデータを入力する際に、それぞれのデータをどのラスタースとして扱うかを選択するセレクターである。

【0043】104は前述のように演算処理結果を格納する為の出力バッファであるが、次段の処理回路105において2ラスタース分のデータを用いることができるように、2個準備している。処理回路105は、リニア変倍処理を行うものである。

【0044】106は演算に際して使用する係数A～Dを与えるレジスタである。

【0045】スキャナーから入力される画像データは、R、G、B各色8ビットで送られてくる。人カマスクブロック101は各色8ビットのデータを画素単位の24ビットにまとめた上で、演算入力バッファ102に格納する。

【0046】図2における横1列のデータ列（すなわち、画像データの1ラインのデータ列）をラスタースと呼ぶ。スキャナーからの画像データは、ラスタース単位で横1列分のデータが左から右に順次転送されてくる。

【0047】演算入力バッファは図1に示すように102-1から102-5の5個が用意されている。それぞれの演算入力バッファは1ラスタース、つまり横1列分の画素データを格納できる容量を持ち、実際にそれぞれの演算入力バッファは各々1ラスタースのデータを格納する。

【0048】5本のラスタース分のデータ入力が行われると、演算入力バッファ102-1から102-5には全てデータが格納された状態となる。本実施例において演算入力バッファが5個用意されているのは 5×5 個のデータが演算に必要なためである。5ラスタースのデータが各々演算入力バッファにセットされるとフィルターブロック103は画像処理演算を開始することが可能となる。

【0049】演算処理の起動管理は入力マスクブロック101がおこなう。入力マスクブロック101は前述のごとく入力データのセットが終了すると、フィルターブロック103に対して起動信号を送り演算をスタートさせる。

【0050】再度図2を用いて、画像データに対する演算の手順について説明する。5個の演算入力バッファにデータがそろった状態は図2の205で示す5ラスタのデータが入力バッファに格納された状態である。

【0051】フィルターブロック103は5個の演算入力バッファ102-1から102-5の各々の中の画面上の一番左側のデータを同時に読み出す。すなわち、画面上で縦1列のデータを同時に読み出す。ついで、その隣りのデータを同時に読み出すというように順次読み出していく。スタート時つまり各ラスタの演算開始時にはこの読み出し動作を連続して5回行うことで25個のデータがセットされる。すなわち、図2の202で囲んだ画素データがセットされる。203は一度に読み出す縦5個のデータを示す。

【0052】5回の読み出し動作の後、フィルターブロック103では、最初の演算を行い結果をアウトバッファ104-1に格納すると、順次203で示す5個（縦*

*1列)のデータを読み出しては演算処理を行い処理結果をアウトバッファ104-1に格納する。この動作を1ラスタについて連続して行う。

【0053】この様にして1ラスタについての処理をおえると、アウトバッファ104-1には204で囲われた範囲1ラインの出力データがそろふ。1ラスタでの処理が終了すると処理を停止する。

【0054】ついで、1ラスタ下のデータから5ラスタを用いて、同様にデータ処理を行い、結果をアウトバッファ104-2に格納するが、この処理に際して、一番上のラスタのデータは演算入力バッファ102-1に格納されており、そのラスタのデータは使用済みであるため不要であるから、このバッファに次のラスタ（すなわち、図2で上から6番目のラスタ）のデータをセットして上記処理を行う。

【0055】このようにして演算入力バッファ102-1から102-5は順に未処理のラスタデータが収められる。フィルターブロックのセクター103-1は演算回路103-2に対してラスタの処理に従って繰り返し

【0056】

【表1】

1st	2nd	3rd	4th	5th
i buff1 → a1x	1 → a5x	1 → a4x	1 → a3x	1 → a2x
i buff2 → a2x	2 → a1x	2 → a5x	2 → a4x	2 → a3x
i buff3 → a3x	3 → a2x	3 → a1x	3 → a5x	3 → a4x
i buff4 → a4x	4 → a3x	4 → a2x	4 → a1x	4 → a5x
i buff5 → a5x	5 → a4x	5 → a3x	5 → a2x	5 → a1x

【0057】の様にデータを入力していく。aの次の数字は縦方向の位置を示す。例えば、a5は5番目すなわち、5×5個のデータの1番下のラスタを示す。また演算結果は、1ラスタ処理毎に2つのアウトバッファに交互に格納する。この様にして画像処理を画像全体に対して行い、最終的に204で囲われた範囲の、処理済み出力データを得る。

【0058】（実施例1特有の効果）このような構成によってn×mのマトリックスデータが必要な演算処理において使用するデータを逐次一回の読み出し動作で使うことが可能となりデータ処理における読み出し時間を短縮し画像処理を効率よく行うことが出来る。

【0059】（実施例2）実施例1では必要なマトリックスデータがn×mの場合にn個の独立に読み出し可能な入力バッファを構成していた。その為、演算処理中はフィルターブロック103がデータの読み出し動作を行っているために、1ラスタ分のデータ処理が終わらないと、次の入力データのセットが出来ない。

【0060】実施例2ではこのバッファをn+1個用意する構成とした。余分に1個の入力バッファを用意することで演算処理の為にフィルターブロック103が読み出し動作を行っている最中にも次の1ラスタのデータを余分の1個の入力バッファにセットすることが可能となる。

【0061】その結果データ処理と入力の並列処理を行うことが出来る様になり、演算処理中に入力を停止する必要はなく、処理時間が短縮できる。

【0062】図5は実施例2のブロック図である。入力バッファ502にセットされたデータの利用方法や演算については実施例と全く同様である。演算入力バッファ502-1～502-6が6個用意されているため、バッファの管理法及び、フィルターブロックへのデータ入力セクターが異なる。

【0063】

【表2】

	1st	2nd	3rd	4th	5th	6th
i buff1	a1x	1	a5x	1	a4x	1
i buff2	a2x	2	a1x	2	a5x	2
i buff3	a3x	3	a2x	3	a1x	3
i buff4	a4x	4	a3x	4	a2x	4
i buff5	a5x	5	a4x	5	a3x	5
i buff6	-	6	a5x	6	a4x	6

【0064】また、後段に対しても同様な理由でアウトバッファ504-1～504-3も1本増設し、演算結果を3個のアウトバッファに順次格納していくようにした。

【0065】（実施例2特有の効果）この構成により、データのセットと処理が同時に可能となる。実施例1においては演算処理実行中は次のデータセットが行えないが、この構成をとることで、演算処理実行中にも次のデータセットが可能となりより効率的なデータ処理が可能となる。

【0066】

【発明の効果】以上説明したように、本発明によれば、

画像データ処理時間を短縮することができる。

【図面の簡単な説明】

【図1】実施例1のブロック図である。

【図2】画素データ配列と処理手順を示す図である。

【図3】演算入力画素配置を示す図である。

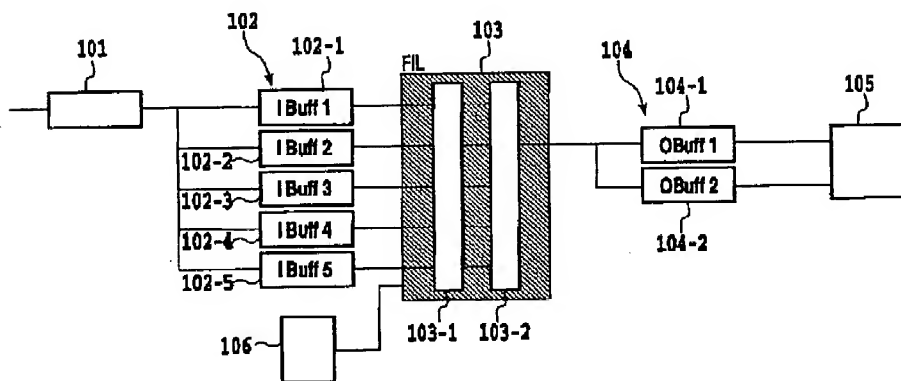
【図4】画素データにおける各色のデータ配置を示す図である。

【図5】実施例2のブロック図である。

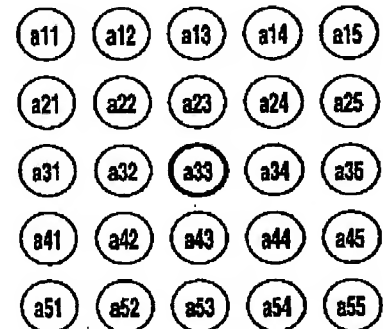
【符号の説明】

- 102 演算入力バッファ
- 103 フィルターブロック
- 104 アウトバッファ

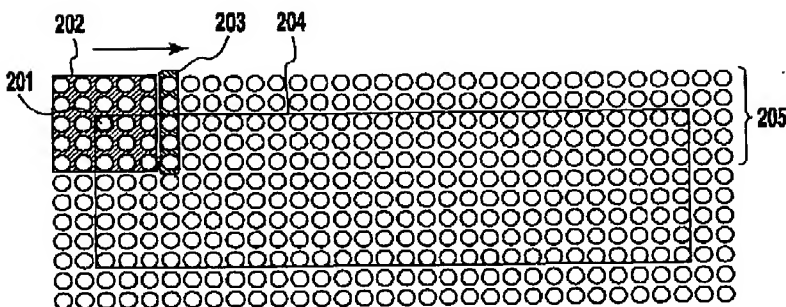
【図1】



【図3】



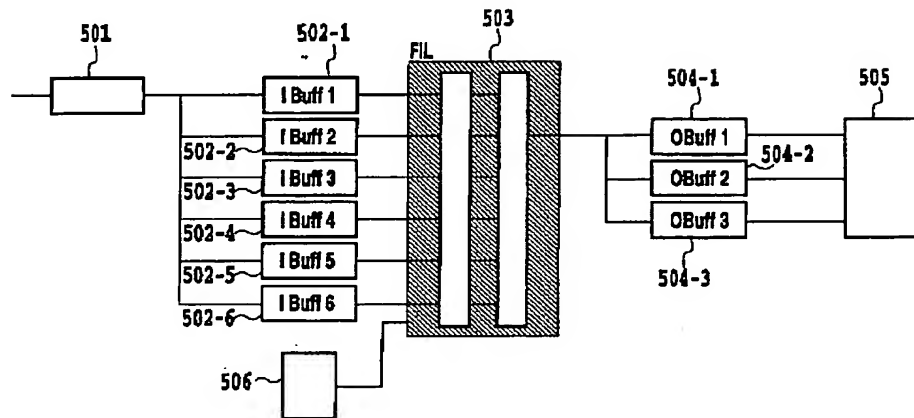
【図2】



【図4】



【図5】



フロントページの続き

(51)Int.Cl. ⁷	識別記号	F I	ターコード(参考)
H 0 4 N 1/21		B 4 1 J 3/12	G 5 C 0 8 2
9/68	1 0 3	G 0 9 G 5/36	5 2 0 C 9 A 0 0 1

Fターム(参考) 2C062 AA24
 5B057 CA01 CA08 CA12 CA16 CB01
 CB08 CB12 CB16 CC02 CE03
 CE05 CH01 CH09 CH11
 5C066 AA11 BA13 CA05 EC02 GA01
 GB01 HA02 KE01 KE09 KE11
 KE16
 5C073 AA01 BB07 BB09 CA02 CE06
 5C077 LL05 LL18 PP03 PP68 PQ12
 PQ24
 5C082 AA27 BA20 CA22 CB01 DA51
 MM02 MM07 MM10
 9A001 HH23 JJ35 KK42

THIS PAGE BLANK (USPTO)